Learning Word Importance with the Neural Bag-of-Words Model

Imran Sheikh[*+], **Irina Illina**[*], **Dominique Fohr**[*], **Georges Linarès**[+]
[*]Université de Lorraine, LORIA, UMR 7503, Vandoeuvre-lès-Nancy, F-54506, France
[*]Inria, Villers-lès-Nancy, F-54600, France
[*]CNRS, LORIA, UMR 7503, Vandoeuvre-lès-Nancy, F-54506, France
[+]Laboratoire Informatique d'Avignon, University of Avignon
{imran.sheikh, irina.illina, dominique.fohr}@loria.fr
georges.linares@univ-avignon.fr

Abstract

The *Neural Bag-of-Words* (NBOW) model performs classification with an average of the input word vectors and achieves an impressive performance. While the NBOW model learns word vectors targeted for the classification task it does not explicitly model *which words are important for given task*. In this paper we propose an improved NBOW model with this ability to learn task specific word importance weights. The word importance weights are learned by introducing a new weighted sum composition of the word vectors. With experiments on standard topic and sentiment classification tasks, we show that (a) our proposed model learns meaningful word importance for a given task (b) our model gives best accuracies among the BOW approaches. We also show that the learned word importance weights are comparable to tf-idf based word weights when used as features in a BOW SVM classifier.

1 Introduction

A Bag-of-Words *BOW* represents text (a sentence, paragraph or a document) as a vector of word features. Traditional BOW methods have used word occurrence frequency and variants of *TermFrequency-InverseDocumentFrequency* (tf-idf) as the word feature (Manning et al., 2008). Development in neural network and deep learning based language processing has led to the development of more powerful continuous vector representation of words (Bengio et al., 2003; Collobert and Weston, 2008; Turian et al., 2010; Mikolov et al., 2013b). It was shown that these predictive word vector representations capture syntactic and/or semantic characteristics of words and their surrounding context (Mikolov et al., 2013a; Pennington et al., 2014), and that they outperform the count based word (vector) representations (Baroni et al., 2014).

Many approaches in text classification are now driven by models built on word vectors. Earlier works proposed models which learned word vectors targeted for the classification task (Maas et al., 2011). In more recent works, text classification is performed with compositional representations learned with neural networks or by training the network specifically for text classification (Goldberg, 2015). One such network is the *Neural Bag-of-Words* (NBOW) model (Kalchbrenner et al., 2014; Iyyer et al., 2015). The NBOW model takes an average of the word vectors in the input text and performs classification with a logistic regression layer. Essentially the NBOW model is a fully connected feed forward network with BOW input. The averaging operation can be attributed to the absence of non-linearity at the hidden layer and the BOW inputs where words are set to 1 (or number of occurrences of that word) and 0. While the NBOW model learns word vectors targeted for the classification task, it does not explicitly model *which words are important for given task*. In this paper, we propose an improved NBOW model which learns these task specific word importance weights. We replace the average with a weighted sum, where the weights applied to each word (vector) are learned during the training of the model. With experiments on sentiment and topic classification tasks, we will show that our proposed model learns meaningful word importance weights and it can perform better than the NBOW model.

The rest of the paper is organised as follows. First in Section 2 we discuss about the related works. In Section 3 we briefly introduce the NBOW model and present our proposed

Proceedings of the 1st Workshop on Representation Learning for NLP, pages 222–229,
Berlin, Germany, August 11th, 2016.

model, termed as the *Neural Bag-of-Weighted-Words* (NBOW2) model, in Section 4. In Section 5 we give details about the experiment setup and the tasks used in our experiments. Section 6 presents a discussion on the word importance weights learned and the classification performance achieved by the proposed NBOW2 model, followed by the conclusion in Section 7.

2 Related Work

A variety of neural network architectures have been proposed for different language processing tasks. In context of text classification, fully connected feed forward networks (Le and Mikolov, 2014; Iyyer et al., 2015; Nam et al., 2014), *Convolutional Neural Networks* (CNN) (Kim, 2014; Johnson and Zhang, 2015; Wang et al., 2015) and also *Recurrent/Recursive Neural Networks* (RNN) (Socher et al., 2013; Hermann and Blunsom, 2013; Dong et al., 2014; Tai et al., 2015; Dai and Le, 2015) have been used. On one hand, the approaches based on CNN and RNN capture rich compositional information, and have been outperforming the state-of-the-art results, on the other hand they are computationally intensive and may require careful hyper-parameter selection and/or regularisation (Zhang and Wallace, 2015; Dai and Le, 2015). We focus our study to the NBOW model which gives an impressive performance in text classification, not far below the state-of-the-art CNN and RNN systems. We propose an improved NBOW model which learns these task specific word importance weights.

Word weights based on variants of word occurrence frequency or tf-idf have been commonly used and studied in literature (Manning et al., 2008; Paltoglou and Thelwall, 2010; Quan et al., 2011). Supervised weighting schemes for adjusting tf-idf for text classification have been proposed earlier (Kim and Zhang, 2014; Deng et al., 2014; Mammadov et al., 2011; Lan et al., 2006). Use of a small number of important words against all the words in the text was studied by Islam(2015) for the task of text relatedness using Latent Semantic Analysis and Google Trigram Model. Our work is in line with recent approaches of text processing with neural networks and learns word importance weights along with the word vectors.

We found that the works by Ling (2015) and Li (2014) are most related to our proposed method. Ling (2015) proposes position based weights to improve word vectors learned by the *Continuous Bag-Of-Words* (CBOW) model (Mikolov et al., 2013a). Li (2014) proposes *Weighted Neural Network* (WNN) for training RNNs which learn compositional representation of text with a parse tree. The WNN weighs how much one specific node contributes to the higher-level representation. Also related are works on learning to pay attention in a sequence of input, as applied in text (Bahdanau et al., 2014) as well as speech (Chan et al., 2015), image (Xu et al., 2015) and protein sequence analysis (Sønderby et al., 2015). The model with convolutional-pooling structures presented by Gao (2014) is also shown to capture keywords in text.

3 Neural Bag-of-Words (NBOW) model

The Neural Bag-of-Words (NBOW) model (Kalchbrenner et al., 2014; Iyyer et al., 2015) is a fully connected network which maps text X, a sequence of words, to one of k output labels. The NBOW model has d dimensional word vectors for each word in the chosen task vocabulary. For the words $w \in X$, corresponding word vectors v_w are looked up and a hidden vector representation z is obtained as an average of the input word vectors

$$z = \frac{1}{|X|} \sum_{w \in X} v_w \tag{1}$$

The average vector z is then fed to a fully connected layer to estimate probabilities for the output labels as:

$$\hat{y} = softmax(W_l\, z + b) \tag{2}$$

where W_l is $k \times d$ matrix, b is a bias vector and $softmax(q) = exp(q)/\sum_{j=1}^{k} exp(q_j)$. For text classification tasks, the NBOW model is trained to minimise the categorical cross-entropy loss (Goldberg, 2015) using a stochastic gradient descent algorithm. Additional fully connected layers can be added into the NBOW model to form Deep Averaging Networks (DAN) (Iyyer et al., 2015).

4 Proposed model: Neural Bag-of-Weighted-Words (NBOW2)

While the NBOW model learns word vectors specialised for the classification task, it lacks to *explicitly* model and provide the information that certain words are more important than the others in

the given classification task. While tf-idf weights capture word importance weights over a given corpus and can be used at the input of the NBOW model, we are interested in letting the model learn the word importance weights which are task specific. We thus propose the NBOW2 model, with the motivation to enable the NBOW model to provide task specific word importance weights.

It is easy to realise that the NBOW model is essentially a fully connected feed forward network with a BOW input vector. The absence of non-linearity at the hidden layer and the BOW inputs where words are set to 1 and 0, results into a sum of the word vectors. However average of word vectors is used as it gives a better performance compared to a sum. To learn the word importance weights, we form a weighted sum composition of the text X as follows:

$$z = \frac{1}{|X|} \sum_{w \in X} \alpha_w \; v_w \tag{3}$$

where α_w are the scalar word importance weights for each word $w \in X$. Learning task specific word vectors with Equation 3 ensures that words which drive the classification task are given higher importance or α_w values (see example in Figure 1).

α_w are obtained by introducing a vector a in the model, and are calculated as follows:

$$\alpha_w = f(v_w \cdot a) \tag{4}$$

where $v_w \cdot a$ represents a dot product between input word vector v_w and vector a; and f scales the importance weights to range $[0, 1]$. Equation 4, which makes the scalar word importance weight α_w a function of the distance of the word w from a in the context space, ensures that calculation of word importance takes into account the contextual word similarities and that it is not biased by the frequency of occurrence of words in the training corpus.

For f common activation functions including softmax, sigmoid and also hyperbolic tangent can be used. From our experiments we found that the sigmoid function $f(t) = (1 + e^{-t})^{-1}$ is a better choice in terms of model convergence and accuracy. However, it must be noted the softmax f could be more interesting in certain tasks because (a) the importance of a word (α_w) in an input document will be dependent not only on the distance of this word from vector a but also on that of the other words in the given input document (b) being a max function the softamax f will bias the composition of input document context vector z (in Equation 3) to only a handful of input words.

To summarise the as compared to the NBOW model, the NBOW2 model will include one additional vector (a). This vector is randomly initialised before training and learned along with the word vectors and other model parameters. The model training, with stochastic gradient descent, and classification (with a forward pass) both will use Equations 3 and 4, along with the output class probability estimates $\hat{y} = softmax(W_l \; z + b)$ similar to the NBOW model.

5 Experiment Setup

To analyse the working and performance of our proposed NBOW2 model, we consider two common tasks: (a) binary sentiment classification on IMDB (Maas et al., 2011) and Rotten Tomatoes movie review dataset (Pang and Lee, 2005) (b) topic classification of 20 Newsgroup dataset. We make available the source code used in our experiments[1].

5.1 Sentiment Analysis

For the IMDB task we use the original dataset[2] with 25000 train and 25000 test movie reviews. For Rotten Tomatoes (RT) we obtained the v1.0 dataset[3] and we do 10-fold cross-validation over the balanced binary dataset of 10,662 sentences. In both IMDB and RT tasks, model training parameters[4] for NBOW2 are kept similar to those chosen for NBOW by Iyyer (2015) after cross validation. For NBOW and NBOW2 models '-RAND' suffix will denote random word vector initialisation and no suffix is initialisation with publicly available 300-d GloVe vectors trained over the Common Crawl (Pennington et al., 2014)[5].

5.2 20 Newsgroup Topic Classification

We use the 'bydate' train/test splits, cleaned and made available by Cardoso (2007)[6]. There are 11,293 documents in the original training set and 7,528 in the test set. For training the NBOW and NBOW2 models, we randomly extract 15%

[1]Source code available at the url https://github.com/mranahmd/nbow2-text-class

[2]http://ai.stanford.edu/ amaas/data/sentiment/

[3]https://www.cs.cornell.edu/people/pabo/movie-review-data/

[4]word vector size 300, word dropout probability 0.3, L2 regularisation weight 1e-5

[5]http://nlp.stanford.edu/projects/glove/

[6]http://web.ist.utl.pt/acardoso/datasets/

of the original train set as the validation set and use remaining 85% as the final training set. Training was performed with the ADADELTA (Zeiler, 2012) gradient descent algorithm. L2 regularisation weight of 1e-5 was applied to all parameters. Further, to add robustness, we applied 75%[7] word dropout (Iyyer et al., 2015; Dai and Le, 2015). Additionally we use early stopping when the validation error starts to increase. Similar to the sentiment analysis experiments '-RAND' suffix will denote random word vector initialisation and no suffix is initialisation with 300-d GloVe.

6 Analysis and Discussion

6.1 Word importance weights learned by the NBOW2 model

We perform an analysis of the word importance weights learned by the NBOW2 model by presenting some qualitative and quantitative results.

6.1.1 Visualisation of word vectors from the RT sentiment analysis task

We visually examine the word vectors learned by the NBOW and NBOW2 models. To visualise word vectors they can be projected into a two dimensional space using the *t-Distributed Stochastic Neighbour Embedding* (t-SNE) technique (van der Maaten and Hinton, 2008). Figure 1 shows the two dimensional t-SNE visualisations of word vectors learned by the NBOW and NBOW2 models. Figure 1a shows a plot of the word vectors learned by the NBOW model and Figure 1b shows a plot of the word vectors learned by the NBOW model. Additionally in Figure 1b each word is given a colour based on the word importance assigned to it by the NBOW2 model.

From Figure 1a we can see that NBOW model tries to separate the words in the word vector space. According to the word examples labelled in Figure 1a the words appear to be grouped into regions corresponding to positive and negative sentiments of the RT movie review task. Similarly the NBOW2 model also learns to separate the words into regions of positive and negative sentiments as shown, by the same word examples, in Figure 1b. If we examine the word importance assigned by the NBOW2 model, indicated by colours in Figure 1b, it is evident that the NBOW2 model also learns to separate words based on their importance weights. To support this statement we show additional word examples labelled in different regions in Figure 1b. For instance the words *a, on, it, for, there* are not so important[8] for the RT sentiment classification task and are present together in region of lowest word importance. The words *staid, inflated, softens* can contribute to (negative) polarity of the reviews and hence have relatively higher importance weights (and are present together near the negative sentiment region in the word vector space).

To further verify our claim that, in comparison to the NBOW model, the NBOW2 model is able to distinguish words based on their importance we show Figure 1c. Figure 1c shows the word vectors learned by the NBOW model (same as in Figure 1a) but it depicts each word with (a colour based on) word importance weight learned by the NBOW2 model. It can be seen in Figure 1c that the NBOW model does not separate/group words based on word importance, even if we restrict only to the example words *a, on, it, for, there*.

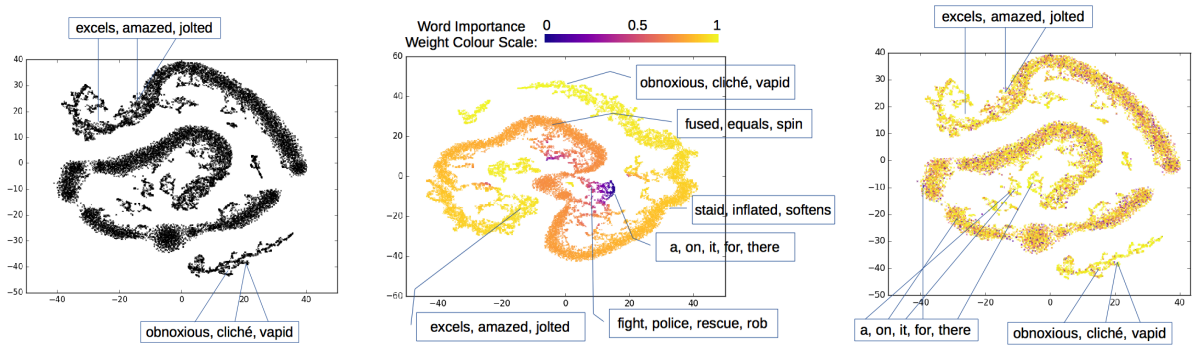
6.1.2 Word importance weights v/s Tf-Idf weights as classification features

In this analysis, we compare the word importance weights learned by the NBOW2 model with tf-idf weights and other word weight features proposed in the previous works. For this comparison, an SVM classifier is used for the IMDB and RT binary classification tasks. Each train/test document is represented as a sparse BOW feature vector in which each word feature is only the word weight. For NBOW2 model it is the scalar word importance weight learned by the model. We compare it with (a) classical tf-idf weights (b) credibility adjusted tf-idf (cred-tf-idf) weights proposed by Kim (2014) (c) binary cosine-normalised weights (bnc) and binary delta-smoothed-idf cosine-normalised (bΔ'c) weights used by Maas (2011) (d) the Naive-Bayes SVM (NBSVM) method proposed by Wang (2012). Tf-idf, bnc and bΔ'c word weights are task independent word weights but cred-tf-idf and NBSVM are built based on the class/task information. It should be noted that some of these methods/features have been the earlier state-of-the-art results for IMDB and RT tasks.

The classification accuracies obtained by the SVM classifiers are reported in Table 1. The tf-

[7] choice based on accuracy on validation set

[8] from a BOW sentiment classification perspective; for other approaches or text analysis they might be essential



(a) (b) (c)

Figure 1: Visualisation of word vectors learned by the NBOW and NBOW2 models in the RT task. Word vectors are reduced to 2 dimension using t-SNE technique and shown in each plot. Plot (a) represents word vectors from the NBOW model, (b) represents words from NBOW2 model, with colours indicating the word importance weights learned by the NBOW2 model, (c) represents word vectors from the NBOW model as in (a) but depicts each word with word importance weight learned by the NBOW2 model.

Features for SVM Classifier	**IMDB**	**RT**
bnc (Maas et al., 2011)	87.8	-
bΔ'c (Maas et al., 2011)	88.2	-
tf-idf-uni (Kim and Zhang, 2014)	88.6	77.1
cred-tf-idf-uni (Kim and Zhang, 2014)	88.8	77.5
NBSVM-uni (Wang and Manning, 2012)	88.3	78.1
NBOW2-RAND Word Importance Weights	88.2	76.7
NBOW2 Word Importance Weights	88.3	76.3

Table 1: Classification accuracy obtained for the IMDB and Rotten Tomatoes (RT) movie reviews sentiment classification task by training an SVM classifier on different word weights as features. (For IMDB 0.1% corresponds to 25 test documents. For RT 1% is about 10 test sentences.)

idf, cred-tf-id and NBSVM methods are denoted with a '-uni' suffix in Table 1 following the notation used by Kim (2014) . For the SVM classifier on 25k full length documents of IMDB task, the NBOW2 model weights are as good as NBSVM and bΔ'c and better than bnc. But they do not perform as good as tf-idf weights. Whereas for the RT task with 1066 test sentences, the NBOW2 model word weights achieve accuracy closer to tf-idf variants.

6.2 NBOW2 model Classification Performance

After the discussion on the word importance weights learnt by the NBOW2 model we compare the classification results obtained with our NBOW2 model. We compare the NBOW2 model classification accuracy to that obtained from the NBOW model (Iyyer et al., 2015),

Model	**IMDB**	**RT**
NBOW-RAND (Iyyer et al., 2015)	88.9	76.2
NBOW (Iyyer et al., 2015)	89.0	79.0
NBOW2-RAND	88.7	78.2
NBOW2	**89.1**	**80.5**
NBSVM-uni (Wang and Manning, 2012)	88.3	78.1
NBSVM-bi (Wang and Manning, 2012)	91.2	79.4
CNN-MC (Kim, 2014)	-	81.1
CNN-non-static (Kim, 2014)	-	**81.5**
s2-bow*n*-CNN (Johnson and Zhang, 2015)	**92.3**	-
SA-LSTM (Dai and Le, 2015)	**92.8**	**83.3**
LM-LSTM (Dai and Le, 2015)	92.4	78.3

Table 2: IMDB and Rotten Tomatoes (RT) movie reviews binary classification accuracy. First group lists BOW methods; including different initialisations of NBOW and NBOW2 (this work). The next group shows best reported results with bi-gram BOW and CNN methods, followed by LSTM RNN. Best method in each group is shown in bold. (For IMDB 0.1% corresponds to 25 test documents. For RT 1% is about 10 test sentences.)

BOW approaches based on Restricted Boltzmann Machines (RBM) and Support Vector Machines (SVM) and more complex approaches based on RNN, CNN. It must be noted that the CNN and RNN based approaches operate on rich word sequence information and have been shown to perform better than BOW approaches on these tasks.

Table 2 compares the classification accuracy of the NBOW2 model on IMDB and Rotten Tomatoes (RT) movie reviews binary classification tasks. Table 3 compares the classification accuracy on 20 Newsgroup topic classification. Results in Table 2 and 3 indicate that the NBOW2 model

Model	Accuracy (%)
NBOW-RAND	83.2
NBOW	83.2
NBOW2-RAND	82.7
NBOW2	**83.4**
RBM-MLP (Dauphin and Bengio, 2013)	79.5
SVM + BoW (Cardoso-Cachopo, 2007)	82.8
SA-LSTM (Dai and Le, 2015)	84.4
LM-LSTM (Dai and Le, 2015)	**84.7**

Table 3: 20 Newsgroup topic classification accuracy. First group lists BOW methods; including different initialisations of NBOW (Iyyer et al., 2015) and NBOW2 (this work). The second group shows best reported results with LSTM RNN. Best method in each group is shown in bold. (0.2% corresponds to about 15 test set documents.)

gives best accuracy among the BOW approaches. For IMDB and newsgroup task, the accuracy of NBOW2 model is closer to that of NBOW (not statistically significant for the 20 Newsgroup). It is also evident that for RT and newsgroup classification, the performance of NBOW2 is not far from CNN and LSTM methods. For further analysis we also trained the NBOW2 model by simply using fixed tf-idf weights in Equation 3. This gave 87.6% and 79.4% accuracy for IMDB and RT task. Thus we can state that the word importance weights of the NBOW2 model are themselves informative.

7 Conclusion and Future Work

We proposed a novel extension to the NBOW model, which enables the model to learn task specific word importance. With experiments and analysis on sentiment and topic classification tasks, we showed that our proposed NBOW2 model learns meaningful word importance weights. We showed that the NBOW2 model gives the best accuracy among the BOW approaches and it can outperform the NBOW model. This motivates us to explore extensions to the model, including (a) class-specific vectors a_q, instead of a single vector a, to obtain class-specific word importance (b) document context specific word importance weights.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March.

Ana Cardoso-Cachopo. 2007. Improving Methods for Single-label Text Categorization. PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. 2015. Listen, attend and spell. *CoRR*, abs/1508.01211.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA. ACM.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3061–3069. Curran Associates, Inc.

Yann Dauphin and Yoshua Bengio. 2013. Stochastic ratio matching of rbms for sparse high-dimensional inputs. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1340–1348. Curran Associates, Inc.

Zhi-Hong Deng, Kun-Hu Luo, and Hong-Liang Yu. 2014. A study of supervised term weighting scheme for sentiment analysis. *Expert Syst. Appl.*, 41(7):3506–3513, June.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 49–54.

Jianfeng Gao, Patrick Pantel, Michael Gamon, Xiaodong He, and Li Deng. 2014. Modeling interestingness with deep neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2–13, Doha, Qatar, October. Association for Computational Linguistics.

Yoav Goldberg. 2015. A primer on neural network models for natural language processing. *CoRR*, abs/1510.00726.

Karl Moritz Hermann and Phil Blunsom. 2013. The Role of Syntax in Vector Space Models of Compositional Semantics. In *Proceedings of ACL*, August.

Aminul Islam, Evangelos Milios, and Vlado Kešelj, 2015. *Text, Speech, and Dialogue: 18th International Conference, TSD 2015, Pilsen,Czech Republic, September 14-17, 2015, Proceedings*, chapter Do Important Words in Bag-of-Words Model of Text Relatedness Help?, pages 569–577. Springer International Publishing, Cham.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China, July. Association for Computational Linguistics.

Rie Johnson and Tong Zhang. 2015. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 103–112, Denver, Colorado, May–June. Association for Computational Linguistics.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, June. Association for Computational Linguistics.

Yoon Kim and Owen Zhang. 2014. Credibility adjusted term frequency: A supervised term weighting scheme for sentiment analysis and text classification. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 79–83, Baltimore, Maryland, June. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.

Man Lan, Chew-Lim Tan, and Hwee-Boon Low. 2006. Proposing a new term weighting scheme for text categorization. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 763–768. AAAI Press.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1188–1196.

Jiwei Li. 2014. Feature weight tuning for recursive neural networks. *CoRR*, abs/1412.3714.

Wang Ling, Yulia Tsvetkov, Silvio Amir, Ramon Fermandez, Chris Dyer, Alan W Black, Isabel Trancoso, and Chu-Cheng Lin. 2015. Not all contexts are created equal: Better word representations with variable attention. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1367–1372, Lisbon, Portugal, September. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 142–150, Stroudsburg, PA, USA. Association for Computational Linguistics.

Musa Mammadov, John Yearwood, and Lei Zhao, 2011. *AI 2010: Advances in Artificial Intelligence: 23rd Australasian Joint Conference, Adelaide, Australia, December 7-10, 2010. Proceedings*, chapter A New Supervised Term Ranking Method for Text Categorization, pages 102–111. Springer Berlin Heidelberg, Berlin, Heidelberg.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, 2008. *Introduction to Information Retrieval*, chapter Scoring, term weighting, and the vector space model. Cambridge University Press, New York, NY, USA.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June. Association for Computational Linguistics.

Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. 2014. Large-scale multi-label text classification - revisiting neural networks. In Toon Calders, Floriana Esposito, Eyke Hüllermeier, and Rosa Meo, editors, *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD-14), Part 2*, volume 8725 of *Lecture Notes in Computer Science*, pages 437–452. Springer Berlin Heidelberg, September.

Georgios Paltoglou and Mike Thelwall. 2010. A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1386–1395, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 115–124, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Xiaojun Quan, Wenyin Liu, and Bite Qiu. 2011. Term weighting schemes for question categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):1009–1021.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Stroudsburg, PA, October. Association for Computational Linguistics.

Søren Kaae Sønderby, Casper Kaae Sønderby, Henrik Nielsen, and Ole Winther, 2015. *Algorithms for Computational Biology: Second International Conference, AlCoB 2015, Mexico City, Mexico, August 4-5, 2015, Proceedings*, chapter Convolutional LSTM Networks for Subcellular Localization of Proteins, pages 68–80. Springer International Publishing, Cham.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July. Association for Computational Linguistics.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Sida Wang and Christopher D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 90–94, Stroudsburg, PA, USA. Association for Computational Linguistics.

Peng Wang, Jiaming Xu, Bo Xu, Chenglin Liu, Heng Zhang, Fangyuan Wang, and Hongwei Hao. 2015. Semantic clustering and convolutional neural network for short text categorization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 352–357, Beijing, China, July. Association for Computational Linguistics.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 2048–2057.

Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701.

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *CoRR*, abs/1510.03820.